

1/6

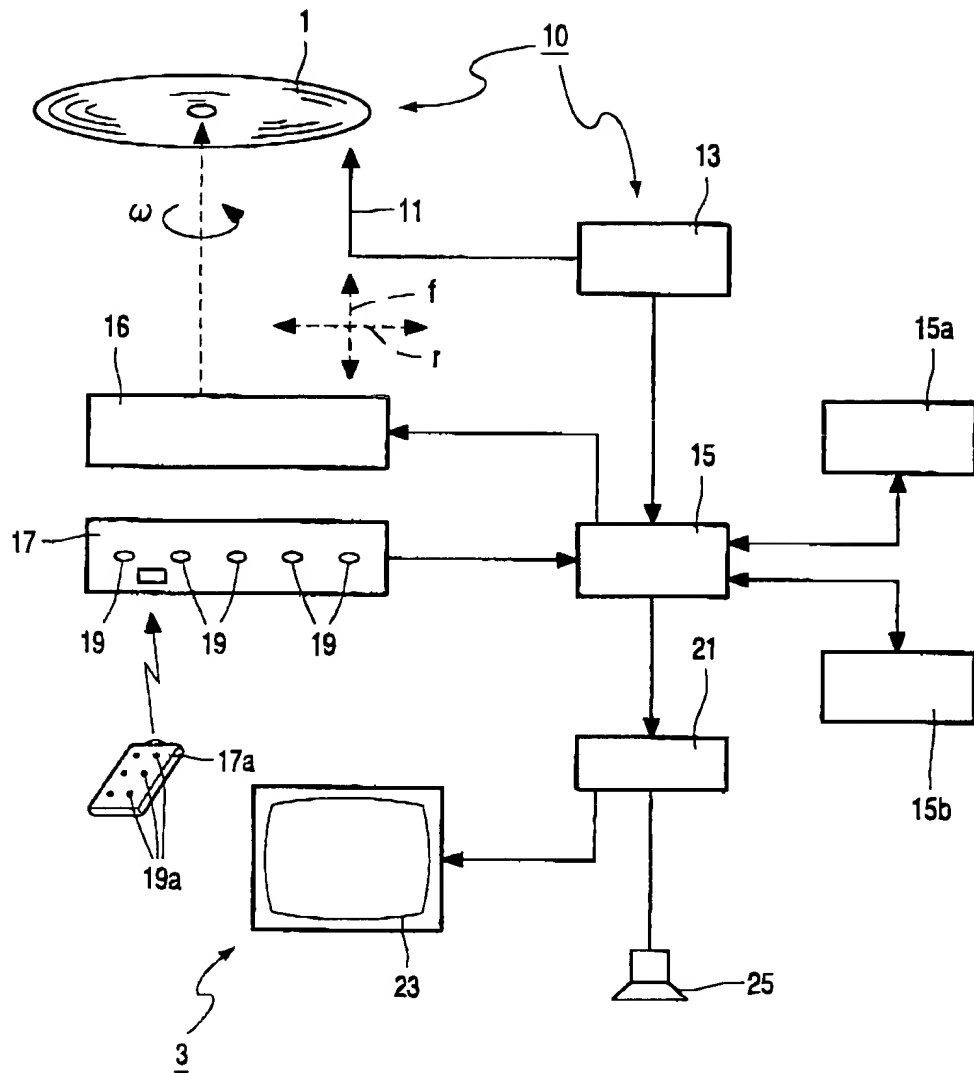


FIG. 1

2/6

Field name	Size (Bits)
Play List Header	8
Number Of Items (NOI)	8
List ID	16
Previous List Offset	16
Next List Offset	16
Return List Offset	16
Playing Time	16
Play Item Wait Time	8
Auto Pause Wait Time	8
Play Item #1 Number	16
.	
.	
Play Item #NOI Number	16

FIG. 2

3/6

Field name	Size (Bits)
Selection List Header	8
Flags	8
Number Of Selections (NOS)	8
Base Of Selection Number (BSN)	8
List ID	16
Previous List Offset	16
Next List Offset	16
Return List Offset	16
Default List Offset	16
Time - out List Offset	16
Wait Time for Time - out	8
Loop Count & Jump Timing	8
Play Item Number	16
Selection #BSN Offset	16
•	
•	
Selection #(BSN + NOS - 1) Offset	16

FIG. 3

4/6

Field name	Size (Bits)
Command List Header	8
Instruction	40
Next_List Offset	16

FIG. 4

Byte #1 0..2 3..4 5..7			Byte #2	Byte #3	Byte #4	Byte #5	Abbreviation and Description	
001	cond0	calc	i	j	k	l	calc	if cond0 { V [j] : = V [k] opcode V [l] }
010	cond0	000	i	--	k	l	move0	if cond0 { V [k] : = V [l] }
	cond1	001	i	j	k	l	move1	if cond1 { V [k] : = V [l] }
	cond0	010	i	j	dd	dd	fill0	if cond0 { V [j] : = dddd }
	00	110	i	j	dd	dd	fillr	while V [i] > = 0 { V [j + V [i]] : = dddd; V [i] -- }
100	cond0	000	i	--	offs	offs	jump0	if cond0 { goto offs }
	cond1	001	i	j	offs	offs	jump1	if cond1 { goto offs }
	cond0	010	i	j	--	--	return	if cond0 { goto V [j] }
	cond0	100	i	--	offs	offs	loop0	if cond0 { dec (V [i]); goto offs }
	cond1	101	i	j	offs	offs	loop1	if cond1 { dec (V [i]); goto offs }
	cond0	110	i	j	offs	offs	loop2	if cond0 { dec (V [i], V [j]); goto offs }
110	cond0	000	i	j	offs	offs	jumpw	if cond0 { wait V [j] deconds for input; goto offs }

FIG. 5

5/6

bit 3..4	Meaning
% 00	True
% 01	$V[i] > 0$
% 10	$V[i] < 0$
% 11	$V[i] = 0$

FIG. 6

bit 3..4	Meaning
% 00	True
% 01	$V[i] > V[j]$
% 10	$V[i] < V[j]$
% 11	$V[i] = V[j]$

FIG. 7

bit 5..7	Meaning
% 000	ADD
% 001	MIN
% 010	MUL
% 011	DIV
% 100	MOD
% 101	AND
% 110	OR
% 111	XOR

FIG. 8

6/6

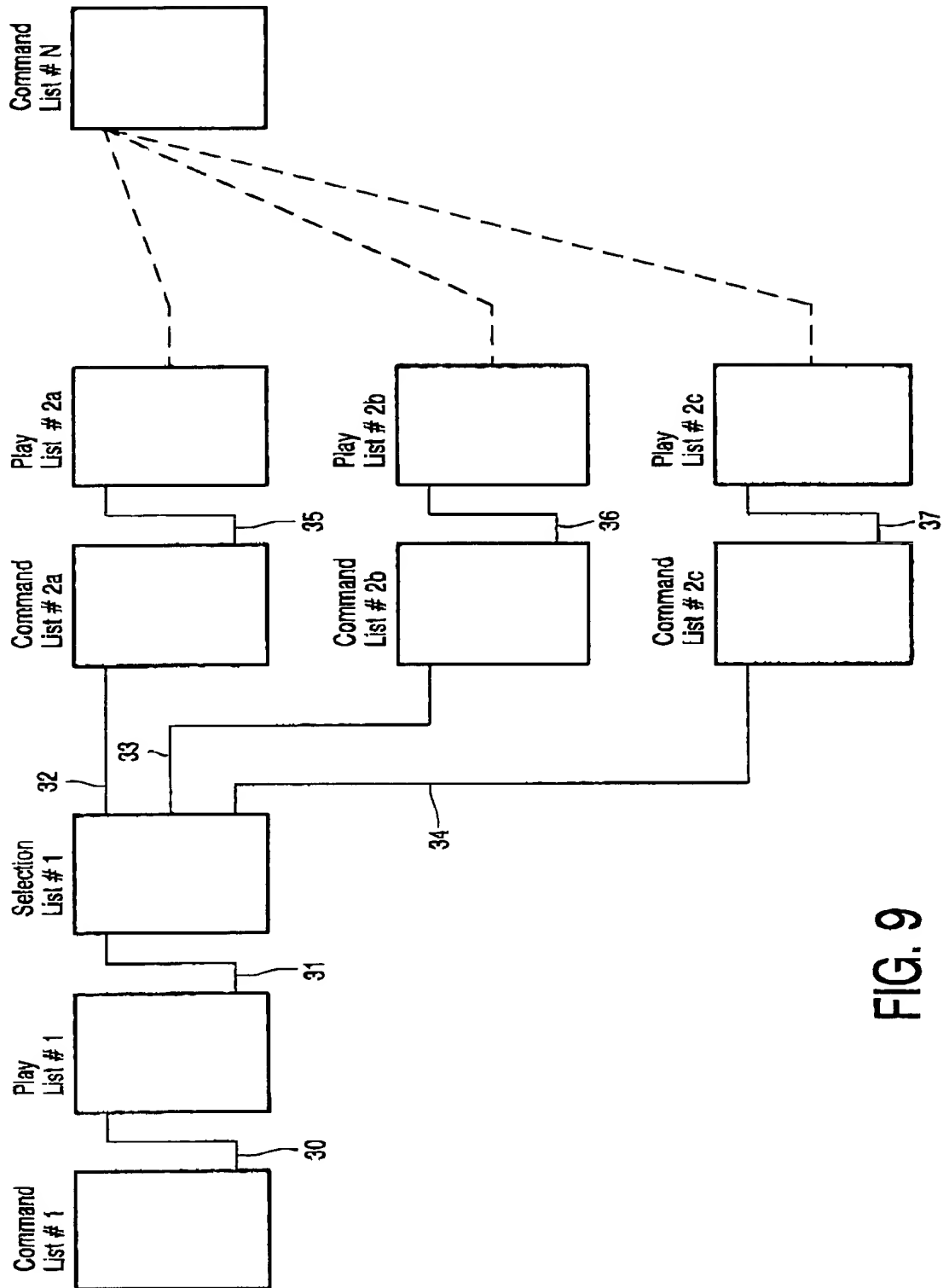


FIG. 9